CONF-8704168--2

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

LA-UR--87-2613

DE87 014742

TITLE THE EVOLUTION OF HOBO

AUTHOR(S) R. A. Clark

SUBMITTED TO CNLS Workshop on Particle Methods in Fluid Dynamics, Los Alamos, NM, April 1987

DISCLAIMER

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

THE EVOLUTION OF HOBO

by R. A. Clark
Los Alamos National Laboratory

I. INTRODUCTION

HOBO is a free-Lagrangian hydrodynamics code which has been under development at Los Alamos for several years. The original version has been described in detail in reference [1], and [2]. It was based on free-Lagrangian tracer points and finite difference approximations to the equations of motion for fluid flow. This method will be reviewed briefly in Section II. The use of an independent time step at each point is described in Section III. This technique increases the speed of the code by factors of 10 to 20. In section IV we will describe how the code has been modified to achieve energy and momentum conservation and how this effects the independent time step. Finally in Section V we will discuss the explicit tracking of material interfaces.

II. BASIC COMPUTATIONAL METHOD

In the original scheme we did the following. We begin with a set of Lagrangian tracer points that essentially replaces a set of Lagrangian zones. Each point has associated with it position, velocity, density, energy, and pressure. At a given time step for a point labeled k choose a representative set of neighbors from which we make a finite difference approximation to the terms $\vec{\nabla}P$ and $\vec{\nabla}\cdot\vec{U}$. We then advance each point from time t to time $t+\delta t$ in accordance with Eqs. (1) through (5).

$$\rho^{n+1} = \rho^n - \delta t \rho \vec{\nabla}\cdot\vec{U} \quad (1)$$

$$\vec{U}^{n+1} = \vec{U}^n - \delta t \vec{\nabla}P/\rho \quad (2)$$

$$e^{n+1} = e^n - P\vec{\nabla}\cdot\vec{U}/\rho \quad (3)$$

$$\vec{x}^{n+1} = \vec{x}^n + \delta t(\vec{U}^{n+1} + \vec{U}^n)/2 \quad (4)$$

$$P^{n+1} = P(\rho^{n+1}, e^{n+1}) \quad (5)$$

Note that we started the cycle by "choosing" a set of representative neighbors. At the next cycle we are free to choose a different set of neighbors for the purpose of approximating $\vec{\nabla}P$ and $\vec{\nabla}\cdot\vec{U}$. However, we do not need to do any rezoning since there really are no zones; the fluid properties are associated with the points. This is what makes the method "free"-Lagrangian. There are many details on how the finite differences are done, artificial viscosity, elastic plastic flow, that can be found in the references.

III. INDEPENDENT TIME STEP

The idea of using an independent time step for each point in the calculation was first suggested to us by Eltgroth [3]. Equations (1) through (3) can be written as

$$\rho^{n+1} = \rho^n + \delta t \dot{\rho} \tag{6}$$

$$u^{n+1} = u^n + \delta t \dot{u} \tag{7}$$

$$e^{n+1} = e^n + \delta t \dot{e} \tag{8}$$

It is the calculation of $\vec{\nabla} P$ and $\vec{\nabla} \cdot \vec{U}$ (which leads to $\dot{\rho}, \dot{u}$, and $\dot{e}$) that takes the vast majority of computation effort, hundreds of floating point operations as opposed to just six to solve Eqs. (6), (7), and (8). The δt in Eqs. (6), (7), and (8) is actually the minimum δt for all points in the mesh taking stability and accuracy into account. At most of the points in the calculation a much larger time step could be used while still maintaining accuracy and stability. Suppose point k could be advanced from t to t + 10 δt in one cycle while maintaining accuracy and stability. Advancing from t to $t + 10\delta t$ in one step is exactly equivalent to advancing in 10 steps of δt each with the values of $\dot{\rho}, \dot{u}$, and $\dot{e}$ frozen. If $\dot{\rho}, \dot{u}$, and $\dot{e}$ are frozen we do not need to calculate $\vec{\nabla} P$ and $\vec{\nabla} \cdot \vec{U}$ at these points and they are for all practical purposes free. In a typical calculation we update $\dot{\rho}, \dot{u}$, and $\dot{e}$ at only 5 to 10% of the points on a typical time step. This is how speedups of 10 to 20 are possible.

One more thing needs to be taken into account. A point may calculate a relatively large time step but then as signals propagate that time step may be reduced. For example, the points ahead of a shock do not know about the approaching shock when there initial time step is calculated. The points are considered either inactive (i.e., frozen) or active. The active points determine the time steps for both themselves and for their inactive neighbors. Hence, a time step is reduced as a shock approaches.

IV. CONSERVATIVE METHOD

The basic numerical method described in Section II makes no effort to conserve momentum and energy in either a local or global sense. This has worked very well on a wide range of problems. However, on calculations involving strong shocks it is simply not adequate. Shock jump conditions can be off by as much as 40%. A conservative method is needed. In Section II we stated that we choose a representative set of points for the finite difference approximation to $\vec{\nabla} P$ and $\vec{\nabla} \cdot \vec{U}$. Actually, we use the neighbors defined by the Vornoi mesh. Some constructs from the Vornoi mesh are shown in Fig. 1, 2, 3, and 4. Figure 1 shows the Vornoi polygon which confines that region of space which is closer to one point than any other point. It can be constructed by drawing the perpendicular bisectors of the lines connecting the point with its neighbors. It is a unique construction. If we draw the connections between each point and its neighbors we have the Delaunay triangles illustrated in Fig. 2 If we draw a polygon around each point whose vertices are the midpoints between itself and each of its neighbors we have Fig. 3. Finally, if we start with Fig. 2 and draw a polygon around the point whose vertices

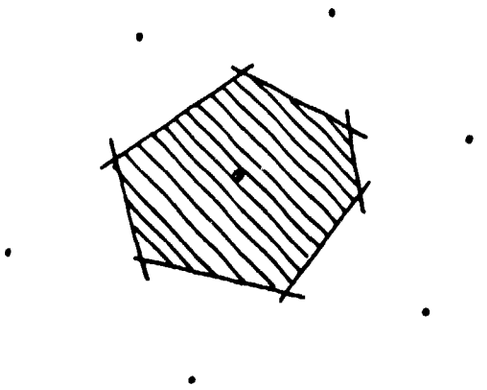

Fig. 1 Vornoi

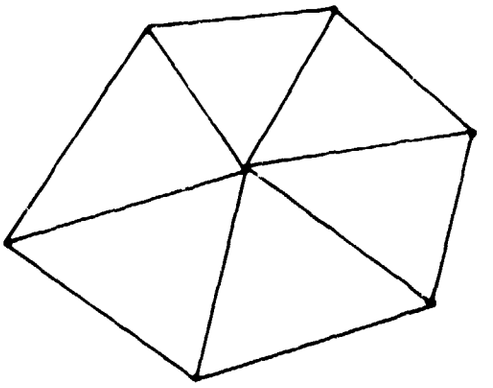

Fig. 2 Delaunay

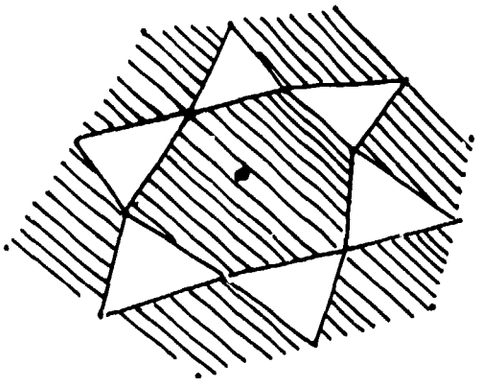

Fig. 3 Midpoints

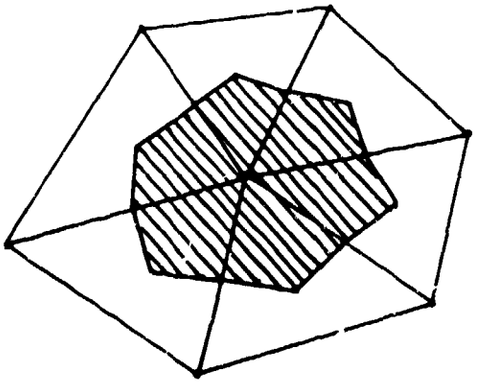

Fig. 4 Median

are the area centroids of the triangles and the midpoints of the connecting lines we get the median mesh, Fig. 4. In order to write a conservative code we want to use a set of zones which connect the entire space of the problem and across whose surfaces we can compute momentum and energy fluxes. In the original nonconservative scheme, although values of P and $\vec{U}$ are well defined at the vertices (by straight averaging), it is unclear how to ever make the method conservative since the zones do not cover all of the space. The Vornoi polygon looks attractive but it is easy to show that the vertices of the Vornoi polygon do not move in a Lagrangian manner so that continuous (and expensive) rezoning would be required. We are reluctant to use the Delaunay triangle because of difficulties that have been observed in other triangle based codes. For our basic computational cell we are using the median mesh of Fig. 4. The vertices move in a Lagrangian manner so that the mass in the zone is constant. Pressure and velocities are integrated over the surface to compute momentum and energy flux. Rezoning is required

whenever neighbors are changed as defined by the motion of the Vornoi mesh. Since we have the values of all of the hydrodynamic variables at the points the rezoning can be done with a high degree of accuracy.

Conservation of energy is assured in the new method by computing the energy flux across each surface and thus ensuring the energy loss of one zone is exactly balanced by the energy gain of another. But when using the independent time step it is possible (in fact, frequent) that one zone is being updated while its neighbor zone is frozen. But if we change the flux rate from A to B while the rate from B to A is frozen we will lose our conservation property. Let us change the way we look at the calculation. Instead of using the cell as the basic computational element we use the surface. The time steps are associated with the surface. The rate of energy flux across a surface may be frozen or updated at each cycle. Now there are two basic types of cells. First, if none of the surfaces surrounding the cell have been updated this cycle then $\dot{\rho}, \dot{e}$ and $\dot{u}$ can be considered frozen, i.e., Eqs. (6), (7), and (8) apply. Second, if any of the surfaces surrounding the zone have been updated the $\dot{\rho}, \dot{e}$ and $\dot{u}$ must be recalculated for the zone. Now everything proceeds as previously described in Section III. Momentum conservation is handled in the same manner. There is some storage cost since there are three times as many surfaces as zones in the problem but the increase in running speed is maintained.

V. Explicit Tracking of material interfaces

The zones defined by the median mesh work extremely well in a single material. However they will always produce a jagged edge at material interfaces and this is undesireable. For this reason we modify the zones at interfaces. The median cell is construced by connecting the midpoints between neighbors with the centroids of the triangles (Figure 5a). If the triangle is made up a two materials we draw a straight line between the midpoints of differing material points and draw a second line from the other midpoint to the opposite vertex. The intersection of these two lines then replaces the triangle centroid in the construction (Figure 5b).

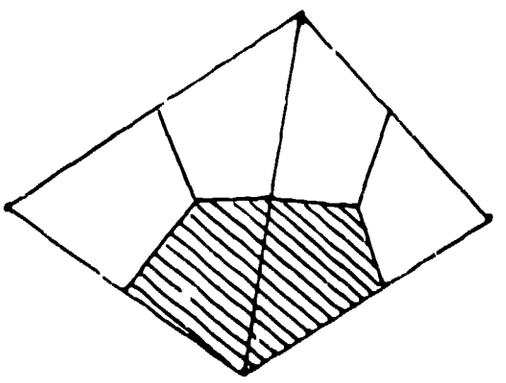

Fig 5a Interior Point

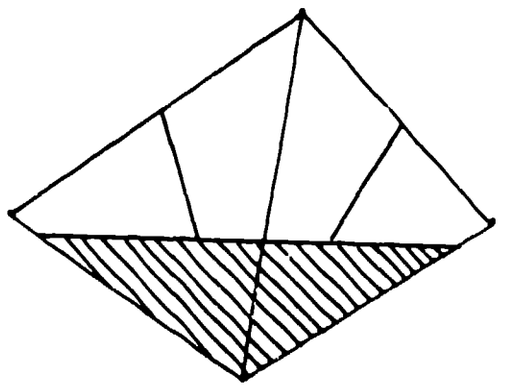

Fig 5b Material Interface

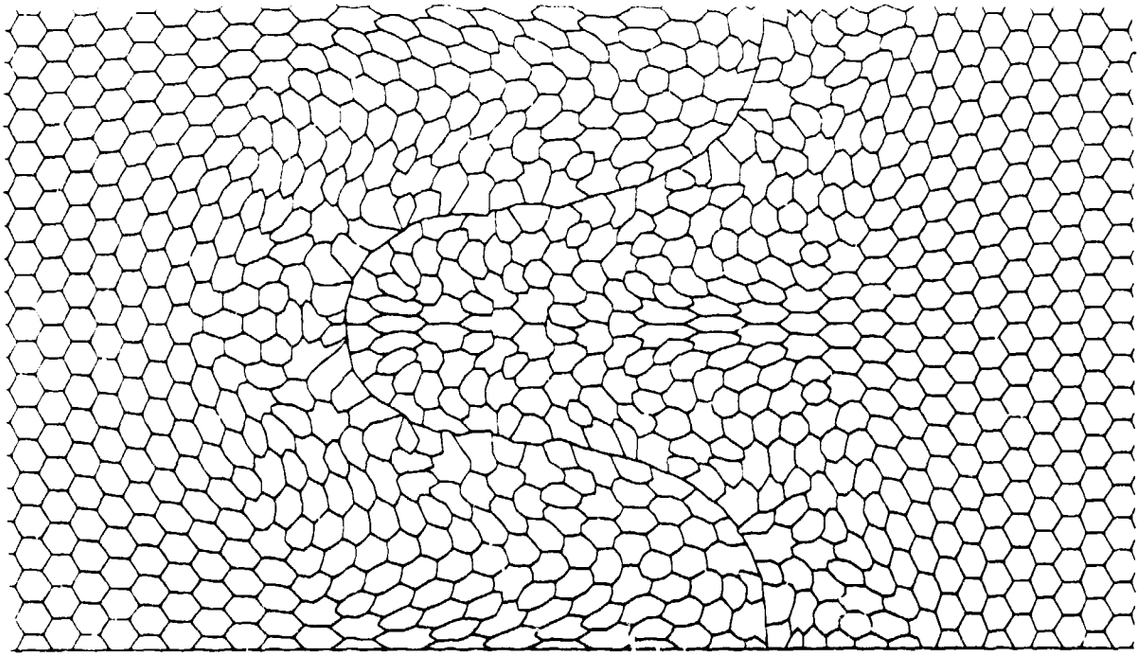

Fig. 6 Typical Mesh

We carry this one step further by allowing the interface intersection with the line connecting points of difference material types to move at a velocity other than the straight average of the two point velocities. We accomplish this by introducing the variable ZED, which is between 0.0 and 1.0, and tells us where the interface intersects between points. A typical resulting mesh is shown in Figure 6.

At the present time we use a Riemann solver to determine the motion of the interface but we are also experimenting with other methods.

REFERENCES

1. R. Clark, "Compressible Lagrangian Hydrodynamics Without Lagrangian Cells," Proceedings of the First International Conference on Free-Lagrangian Methods, Lecture Notes in Physics, Vol. 238, (Springer Verlag, New York 1985), pp. 281-294.

2. R. Clark, "Free Lagrangian Hydrodynamics Using Massless Tracer Points," Proceedings of the Tenth International Conference On Numerical Methods in Fluid Dynamics, June 23-27, Beijing, China, (1986) to be published.

3. P. G. Eltgroth, "Free Lagrangian Methods, Independent Time Steps, and Parallel Processing," Proceedings of the First International Conference on Free-Lagrangian Methods, Lecture Notes in Physics, Vol. 238, (Springer Verlag, New York, 1985), pp. 114-121.